

# **Разработка технического задания**

# Определения

- **Автоматизированная система (АС)** – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.
- **Алгоритм функционирования автоматизированной системы** – алгоритм, задающий условия и последовательность действий компонентов автоматизированной системы при выполнении ею своих функций.

- ***Взаимодействие автоматизированных систем*** – обмен данными, командами и сигналами между функционирующими АС.
- ***Документация на автоматизированную систему*** – комплект взаимосвязанных документов, полностью определяющих технические требования к АС, проектные и организационные решения по созданию и функционированию АС.

- ***Жизненный цикл автоматизированной системы***  
– совокупность взаимосвязанных процессов создания и последовательного изменения состояния АС от формирования исходных требований к ней до окончания эксплуатации и утилизации комплекса средств автоматизации АС.
- ***Интегрированная автоматизированная система***  
– совокупность двух или более взаимоувязанных АС, в которой функционирование одной из них зависит от результатов функционирования другой (других) так, что эту совокупность можно рассматривать как единую АС.

- ***Информационная система*** – система, которая организует хранение и манипулирование информацией о предметной области.
  
- ***Информационная совместимость автоматизированных систем*** – частная совместимость АС, характеризуемая возможностью использования в них одних и тех же данных и обмена данными между ними.

- **Информационное обеспечение автоматизированной системы** – совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации, применяемой в АС при ее функционировании.

- **Методическое обеспечение автоматизированной системы** – совокупность документов, описывающих технологию функционирования АС, методы выбора и применения пользователями технологических приемов для получения конкретных результатов при функционировании АС.
- **Организационное обеспечение автоматизированной системы** – совокупность документов, устанавливающих организационную структуру, права и обязанности пользователей и эксплуатационного персонала АС в условиях функционирования, проверки и обеспечения работоспособности АС.

- ***Приемочная документация на автома-тизированную систему*** – документация, фиксирующая сведения, подтверждающие готовность АС к приемке ее в эксплуатацию, соответствие АС требованиям нормативных документов.
- ***Программная совместимость автома-тизированных систем*** – частная совместимость АС, характеризуемая возможностью работы программ одной системы в другой и обмена программами, необходимыми при взаимодействии АС.

- **Процесс создания автоматизированной системы** – совокупность работ от формирования исходных требований к системе до ввода в действие.
- **Рабочая документация на автоматизированную систему** – комплект проектных документов на АС, содержащий взаимоувязанные решения по системе в целом, ее функциям, всем видам обеспечения АС, достаточные для комплектации, монтажа, наладки и функционирования АС, ее проверки и обеспечения работоспособности.

- ***Совместимость автоматизированных систем*** – комплексное свойство двух или более АС, характеризующее их способностью взаимодействовать при функционировании.
- ***Сообщение автоматизированной системы*** – сведения в виде законченного блока данных, передаваемые при функционировании АС.

- ***Спецификация программы*** – формализованное представление требований, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также описание задачи, условия и эффекта действия без указания способа его достижения.

- ***Технический проект автоматизированной системы*** – комплект проектных документов на АС, разрабатываемый на стадии «Технический проект», утвержденный в установленном порядке, содержащий основные проектные решения по системе в целом, ее функциям и всем видам обеспечения АС и достаточный для разработки рабочей документации на АС.

- ***Техническое задание на авто-матизированную систему*** – документ, оформленный в установленном порядке и определяющий цели создания АС, требования к АС и основные исходные данные, необходимые для ее разработки, а также план-график создания АС.
- ***Техническое обеспечение автоматизированной системы*** – совокупность всех технических средств, используемых при функционировании АС.

- *Эксплуатационная документация на автоматизированную систему* – часть рабочей документации на АС, предназначенная для использования при эксплуатации системы, определяющая правила действия персонала и пользователей системы при ее функционировании, проверке и обеспечении ее работоспособности.

# Основные стандарты, методологии и своды знаний, где упоминается ТЗ или SRS (Software (or System) Requirements Specification):

- ГОСТ 34
- ГОСТ 19
- IEEE STD 830-1998
- ISO/IEC/ IEEE 29148-2011
- RUP
- SWEBOK, BABOK и пр.

Спецификация требований программного обеспечения ([англ.](#) Software Requirements Specification, SRS) — структурированный набор требований (функционал, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам. (Определение на основе IEEE Std 1012:2004) Предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком (или подрядчиками) о том, как должен функционировать программный продукт.

Как инструмент коммуникации в связке общения заказчик-исполнитель, техническое задание позволяет:

### 1. обеим сторонам:

- представить готовый продукт;
- выполнить по пунктной проверке готового продукта (приемочное тестирование — проведение испытаний);
- уменьшить число ошибок, связанных с изменением требований в результате их неполноты или ошибочности (на всех стадиях и этапах создания, за исключением испытаний);

### 2. заказчику:

- осознать, что именно ему нужно
- требовать от исполнителя соответствия продукта всем условиям, оговоренным в ТЗ

### 3. исполнителю:

- понять суть задачи, показать заказчику «технический облик» программного изделия или автоматизированной системы;
- спланировать выполнение проекта и работать по намеченному плану – отказаться от выполнения работ, не указанных в ТЗ.

# Когда нет продуманного и четкого технического задания



What the customer said  
that they wanted



How the Sales Rep  
understood it



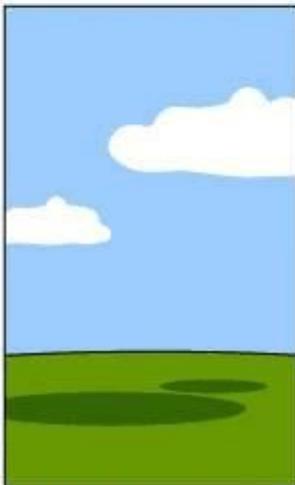
How Solution Mngmnt  
wrote the requirements



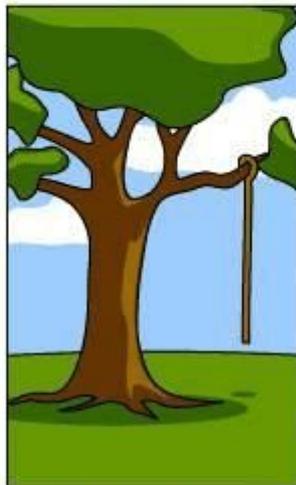
How the Developers  
coded it



How Marketing  
described it



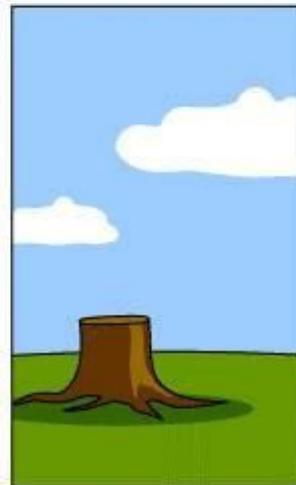
How the project was  
documented



What Services  
implemented



How the Customer was  
billed



How it was supported



What the Customer  
really needed

## Вопрос: *Как разработать техническое задание?*

- *Коммерческая организация* решила внедрить у себя автоматизированную систему. Она *не имеет собственной ИТ-службы* и решили поступить так: заинтересованное лицо должно разработать Техническое задание и отдать его на разработку сторонней организации.
- *Коммерческая организация* решила внедрить у себя автоматизированную систему. Она *имеет собственную ИТ-службу*. Решили поступить так: разработать Техническое задание, затем согласовать его между ИТ-службой и заинтересованными лицами, и реализовать собственными силами.
- *Госструктура решила затеять ИТ-проект*. Тут все настолько мутно, куча формальностей, откатов, распилов и пр.

# Вопрос: *Как разработать техническое задание?*

IT-компания занимается услугами по разработке и/или внедрению автоматизированных систем. Это наиболее сложный случай, ведь приходится работать в самых различных условиях:

- Техническое задание разрабатывается для собственных разработчиков (клиенту все равно);
- Техническое задание разрабатывается для передачи подрядчику (т.е. группе программистов, находящихся за штатом компании, или отдельному специалисту).

# Что такое техническое задание?

ГОСТ: «ТЗ на АС является основным документом, определяющим требования и порядок создания (развития или модернизации — далее создания) автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка при вводе в действие»

# Что такое техническое задание?

«Техническое задание – это исходный документ на проектирование технического объекта.

Техническое задание устанавливает основное назначение разрабатываемого объекта, его технические и тактико-технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.

- ГОСТ 2.114-95 Единая система конструкторской документации. Технические условия
- ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению
- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы

***Основное назначение Технического задания —***  
сформулировать требования к разрабатываемому объекту, т.е. к автоматизированной системе

***Требования по ГОСТ к АС:***

- требования в функциональности - ***90% сложности***
- требования к безопасности и правам доступа;
- требования к квалификации персонала и т.п.

Виды требований могут быть различными – это зависит от целей проекта

***Свойства требований:***

- Требование должно быть **понятным**;
- Требование должно быть **конкретным**;
- Требование должно быть **тестируемым**.

## **ВАЖНО!!!**

- На каком языке (в смысле сложности понимания) должно быть написано техническое задание?
- Должны ли быть описаны в нем спецификации различных функций, алгоритмы, типы данных и прочие технические штуки?
- А что такое техническое проектирование (указано в ГОСТах), и как оно связано с Техническим заданием?
- Где граница между Техническим заданием и Техническим проектом?

**Техническое задание** – это документ, в основе которого лежат требования, сформулированные *на понятном (обычном, привычном) для Заказчика языке.*

Используется отраслевая терминология, понятная Заказчику.

Никаких привязок к особенностям технической реализации быть не должно. Т.е. на этапе ТЗ в принципе не важно, на какой платформе будут реализовываться эти требования.

Исключения: Если речь идет о внедрении системы на основе уже существующего программного продукта (привязка на уровне экранных форм, форм отчетов и пр.)

Выяснением и формулированием требований, а также разработкой Технического задания должен заниматься **бизнес-аналитик.**

**Технический проект** – это документ, который предназначен для технической реализации требований, сформулированных в Техническом задании.

В этом документе описываются структуры данных, триггеры и хранимые процедуры, алгоритмы и прочие штуки, которые потребуются **техническим специалистам**.

Технический проект делает **Архитектор системы** (вот совмещение этой роли с программистом вполне нормально), а точнее группа специалистов АО главе с архитектором.

**!!!** Чем больше проект, тем и больше людей работает над Техническим заданием.

# **ЗАДАЧА:**

Сделать качественное Техническое задание, понятное Заказчику, а Технический проект использовать как внутренний документ для взаимоотношений между архитектором системы и программистами.

# Структура технического задания по ГОСТ:

- общие сведения;
- назначение и цели создания (развития) системы;
- характеристика объектов автоматизации;
- требования к системе;
- состав и содержание работ по созданию системы;
- порядок контроля и приемки системы;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- требования к документированию;
- источники разработки.

# Техзадание. Раздел 1. Общие сведения: начало

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
полное наименование системы и ее условное обозначение	Пишем, как будет называться система, ее краткое наименование
шифр темы или шифр (номер) договора	Это не актуально, но можно и указать, если требуется
наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты	указывают, кто (какие организации) будут работать над проектом. Можно указать и их роли. Можно вообще удалить этот раздел (достаточно формальный).
перечень документов, на основании которых создается система, кем и когда утверждены эти документы	Полезная информация. Здесь стоит указать ту нормативно-справочную документацию, которую Вам предоставили для ознакомления с определенной частью требований

# Техзадание. Раздел 1. Общие сведения: продолжение

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
плановые сроки начала и окончания работы по созданию системы	Пожелания по срокам. Иногда в ТЗ об этом пишут, но чаще такие вещи описываются в договорах на работы
сведения об источниках и порядке финансирования работ	Аналогично, как и в предыдущем пункте про сроки. Более актуально для государственных заказов (для бюджетников)
порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы	Пункт может быть не описан, т.к. требования к документированию вынесены отдельно, а кроме этого есть целый отдельный раздел «Порядок контроля и приемки» системы

# Техзадание. Раздел 2. Назначение и цели создания (развития) системы: начало

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
Назначение системы	<p>С одной стороны с назначением все просто. Но желательно формулировать конкретно. Если написать что-то вроде «качественно автоматизировать складской учет в компании X», то потом можно долго обсуждать результат при его завершении, даже независимо от хорошей формулировки требований, т.к. Заказчик всегда может говорить, что под качеством он имел в виду нечто иное.</p> <p>Лучше сразу написать примерно так: «Система предназначена для ведения складского учета в компании X в соответствии с требованиями, зафиксированными в данном Техническом задании».</p>

# Техзадание. Раздел 2. Назначение и цели создания (развития) системы: продолжение

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
Цели создания системы	<p>Цели – это безусловно важный раздел. Если уж его включать, то надо уметь эти цели формулировать. <i>Пример неудачной цели:</i> «Обеспечить быстрое оформление документов менеджером». Что такое быстрое? Это можно потом доказывать бесконечно.</p> <p>Если это важно, то лучше переформулировать данную цель так: «Менеджер по продажам должен иметь возможность оформить документ «Реализация товаров» из 100 строк за 10 минут».</p> <p>Подобная цель может появиться, если, например, в настоящее время менеджер тратит на это около часа, что слишком много для этой компании и для них это важно.</p>

# Техзадание. Раздел 3. Характеристика объектов автоматизации :

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию	На практике обычно это не включают. Но можно привести ссылки на документы, которые полезно изучить составу проектной команды для погружения в вопрос (отраслевые особенности, например)
сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды	Не актуально для проектов по автоматизации учета

# Техзадание. Раздел 4. Требования к системе :

*ГОСТ расшифровывает перечень таких требований:*

- требования к структуре и функционированию системы;
- требования к численности и квалификации персонала системы и режиму его работы;
- показатели назначения;
- требования к надежности;
- требования безопасности;
- требования к эргономике и технической эстетике;
- требования к транспортабельности для подвижных АС;
- требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
- требования к защите информации от несанкционированного доступа;
- требования по сохранности информации при авариях;
- требования к защите от влияния внешних воздействий;
- требования к патентной чистоте;
- требования по стандартизации и унификации;

# Техзадание. Раздел 4. Требования к системе :

## **ВАЖНО:**

**Требования к квалификации.** Если квалификация имеющегося персонала явно недостаточна, лучше прописать требования к организации обучения, программе обучения, срокам и т.п.

**Требования к защите информации от несанкционированного доступа.** Это требования к разграничению доступа к данным. Если такие требования планируются, то их нужно расписать отдельно, как можно более детально по тем же правилам, что и функциональные требования (понятность, конкретность, тестируемость).

**Требования к стандартизации.** Такие требования обычно инициирует ИТ-служба Заказчика. Например, у компании 1С есть требования к оформлению программного кода, проектированию интерфейса и пр.

# Техзадание. Раздел 4. Требования к системе :

## ***ВАЖНО:***

***Требования к структуре и функционированию системы.*** Могут быть описаны требования к интеграции систем между собой, представлено описание общей архитектуры. Чаще требования к интеграции выделяют вообще в отдельный раздел или даже отдельное Техническое задание, т.к. эти требования могут оказаться достаточно сложными.

***Требования к видам обеспечения.*** ГОСТ выделяет такие виды:

- Математическое
- Информационное
- Лингвистическое
- Программное
- Техническое
- Метрологическое
- Организационное
- Методическое

# Техзадание. Раздел 5. Состав и содержание работ по созданию системы :

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
<p>Перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ 24.601, сроки их выполнения, перечень организаций — исполнителей работ, ссылки на документы, подтверждающие согласие этих организаций на участие в создании системы, или запись, определяющую ответственного (заказчик или разработчик) за проведение этих работ</p>	<p>Это план разработки системы, ее этапность, возможность привлечения подрядчиков и т.п.</p>

# Техзадание. Раздел 6. Порядок контроля и приемки системы:

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
Виды, состав, объем и методы испытаний системы и ее составных частей (виды испытаний в соответствии с действующими нормами, распространяющимися на разрабатываемую систему)	Правильно выделить этапы работ, способы проверки результатов по этим этапам. Выделить тестируемые требования, понятные Заказчику

# Техзадание. Раздел 7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие:

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
Приведение поступающей в систему информации (в соответствии с требованиями к информационному и лингвистическому обеспечению) к виду, пригодному для обработки с помощью ЭВМ	Весьма важный момент. К примеру, для функционирования системы так, как задумано, может потребоваться использование каких-либо отраслевых или общероссийских справочников и классификаторов. Эти справочники должны каким-то образом появляться в системе, обновляться и правильно использоваться

# Техзадание. Раздел 8. Требования к документированию:

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
Согласованный разработчиком и Заказчиком системы перечень подлежащих разработке комплектов и видов документов	Наличие полноценной документации – важная часть результата. Необходимо заранее оговорить с Заказчиком, какие виды документации будут разрабатываться, как они будут выглядеть (содержание и желательно примеры)

# Техзадание. Раздел 9. Источники разработки:

<i>Рекомендации по ГОСТ</i>	<i>Что с этим делать на практике</i>
<p>Должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы</p>	<p><b><i>Если честно, это ближе к лирике.</i></b> Особенно, когда говорят об экономическом эффекте и прочих вещах, которые объективно посчитать практически невозможно. Т.е. можно конечно, но это будет скорее на бумаге, чисто теоретически</p>

## ***ВАЖНО!!!***

Разделы 1-9 «Могут» быть включены в Техническое задание, но не «Обязаны».

Техзадание должно разрабатываться для достижения результата.

Поэтому, если для Заказчика-Разработчика очевидно, что какой-то отдельный раздел к результату не приблизит, значит он не нужен и не надо тратить на него время

## Раздел 4 «Требования к системе»

**Помним:** требования должны быть понятными, конкретными и тестируемыми!

*Рекомендации из книги К.Вигерс «Разработка требований к программному обеспечению»*

- Не следует использовать слов, имеющих множество синонимов. Если это необходимо, то лучше дать четкое определение термину в разделе «Термины и определения» к Техническому заданию
- Следует стараться не использовать длинных предложений
- Если какое-то требование Вам кажется слишком общим, его необходимо детализировать до более мелких, но конкретных требований

*Рекомендации из книги К.Вигерс «Разработка требований к программному обеспечению» продолжение*

- Используйте больше схем, графиков, таблиц, рисунков – так информацию воспринимается гораздо легче
- Следует избегать таких слов: «эффективный», «адекватный», «простой», «понятный», «быстрый», «гибкий», «улучшенный», «оптимальный», «прозрачный», «устойчивый», «достаточный», «дружественный», «легкий» и др.

# Техническое задание. Виды работ при сборе требований к ИС и информации для описания бизнес-процессов



Решение принято, проекту быть!



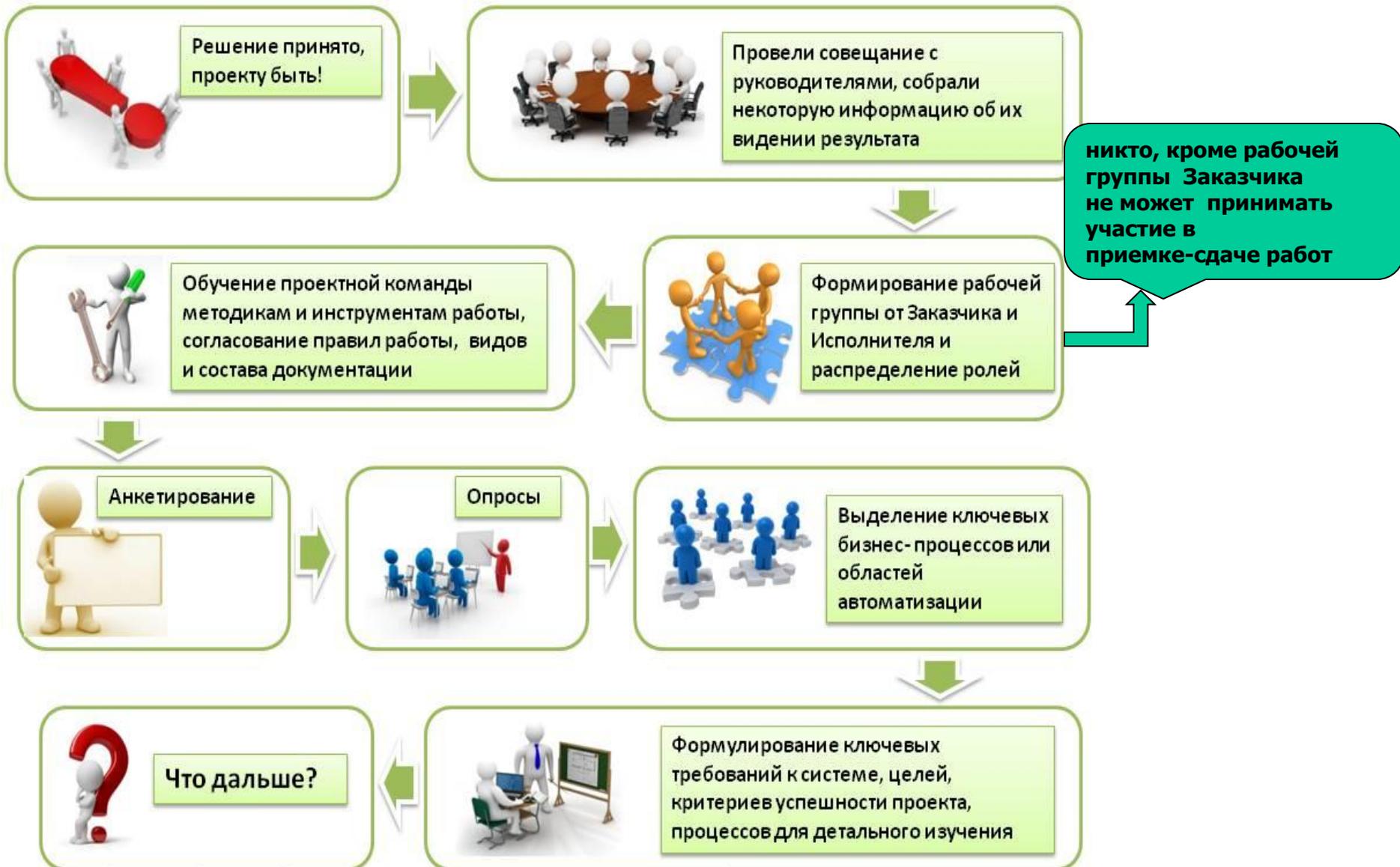
Провели совещание с руководителями, собрали некоторую информацию об их видении результата



Пребывая в уверенности, что консультантам все понятно и им лучше известно, как все должны работать, идем документировать

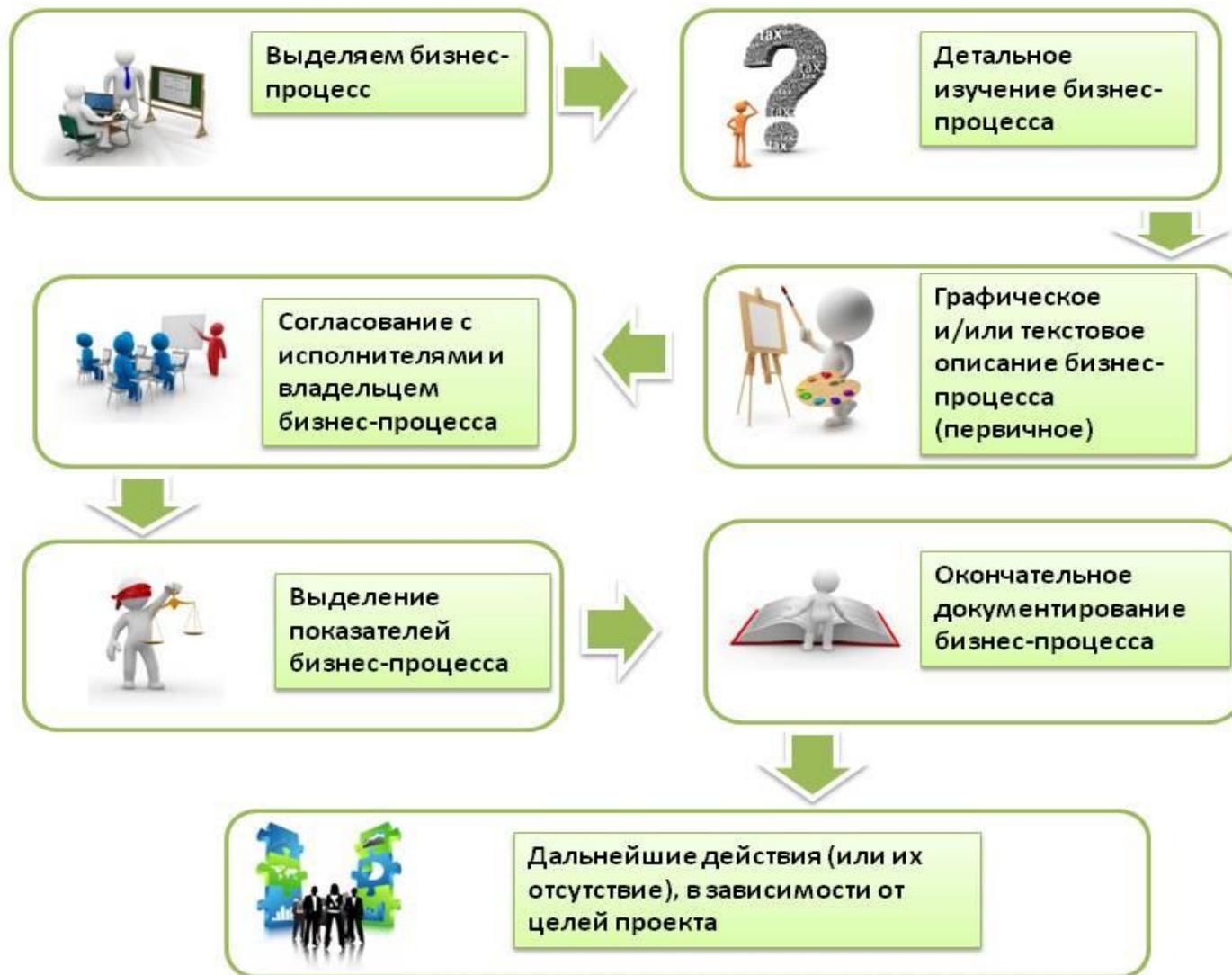
Сложившаяся практика большинства проектов

# Техническое задание. Виды работ при сборе требований к ИС и информации для описания бизнес-процессов



Сбор требований для автоматизации или описания бизнес-процессов

# Техническое задание. Виды работ при сборе требований к ИС и информации для описания бизнес-процессов



Последовательность работ при описании бизнес-процессов

# Подход к изучению требований к информационной системе с дальнейшим их отражением в Техническом задании



Последовательность работ при разработке Технического задания

# **ВАЖНО!!!**

Технического задания процесс трудоемкий, а значит и затратный.

Но если он сделан грамотно, то избавляет Заказчика от несбывшихся ожиданий.

Исполнителю приходится делать то, что требуется Заказчику и не переделывать сто раз одно и то же. Да и вообще, придает всему проекту прозрачность.

# Понятие «стандарт»

- **Стандарт** - *в широком смысле слова образец, эталон, модель,* принимаемые за исходную для сопоставления с ними других подобных объектов
- **Стандарт** как нормативно-технический документ устанавливает комплекс норм, правил, требований к объекту стандартизации

# **Стандарт ISO/IEC 12207 – процессы ЖЦ**

# Основные определения

- **Жизненный цикл ИС** – период времени, который начинается с момента принятия решения о необходимости создания ИС и заканчивается в момент ее полного изъятия из эксплуатации

# Общая модель ЖЦ системы

концепция идеи  
системы

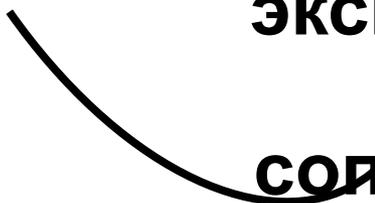
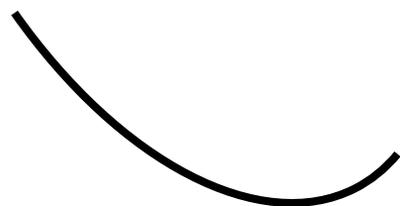
разработк  
а

создани  
е

эксплуатация  
и

сопровожден  
ие

утилизаци  
я



# Характеристика стандарта

- **Область применения стандарта:**  
процессы, выполняющиеся в ходе жизненного цикла программной системы
- **Модель жизненного цикла:**  
структура, состоящую из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта

# Цель стандарта

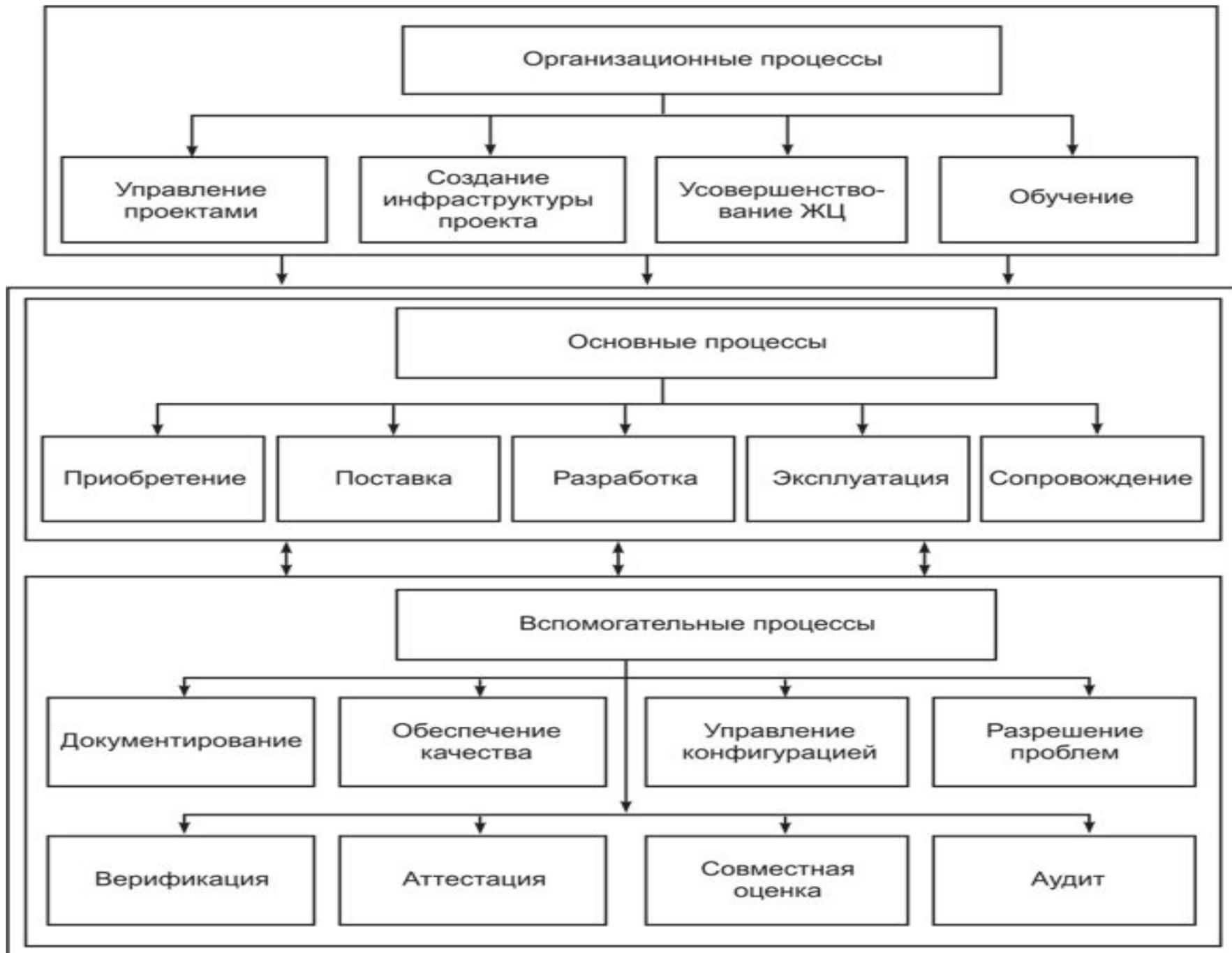
***Определить полную совокупность процессов, которые могут выполняться в ходе проекта по созданию программной системы***

# Адаптация

Поскольку проекты могут сильно различаться, например по масштабам, сложности, рискам и т. п., допускается для каждого проекта локально видоизменять использующиеся в нем процессы, исключая или добавляя отдельные работы и задачи. Такая деятельность называется в стандарте ***адаптацией***

# Группы процессов

- **Основные** - это процессы, непосредственно относящиеся к жизненному циклу информационной системы
- **Вспомогательные** - это процессы, предназначенные для поддержки основных процессов
- **Организационные** - это общекорпоративные процессы, такие как «Обучение» или «Управление»



# **Модель жизненного цикла программного продукта**

- **Под моделью жизненного цикла понимается** структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.

# Состав модели ЖЦ ПО

- **стадии** - часть процесса создания ПО, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта
- **результаты** выполнения работ на каждой стадии
- **ключевые события** - точки завершения работ и принятия решений

# Основные модели ЖЦ

- Каскадная модель
- V-образная модель
- Модель прототипирования
- Модель быстрой разработки приложений
- Многопроходная модель
- Спиральная модель

# Каскадная модель

*Прямолинейная и простая в использовании. Необходим постоянный жесткий контроль за ходом работы. Разрабатываемое программное обеспечение недоступно для изменений*



# V-образная модель

*Простая в использовании. Особое значение придается тестированию и сравнению результатов фаз тестирования и проектирования*



# Модель прототипирования



# Модель быстрой разработки приложений

*Проектные группы небольшие (3...7 человек) и составлены из высококвалифицированных специалистов. Уменьшенное время цикла разработки (до 3 мес) и улучшенная производительность. Повторное использование кода и автоматизация процесса разработки*

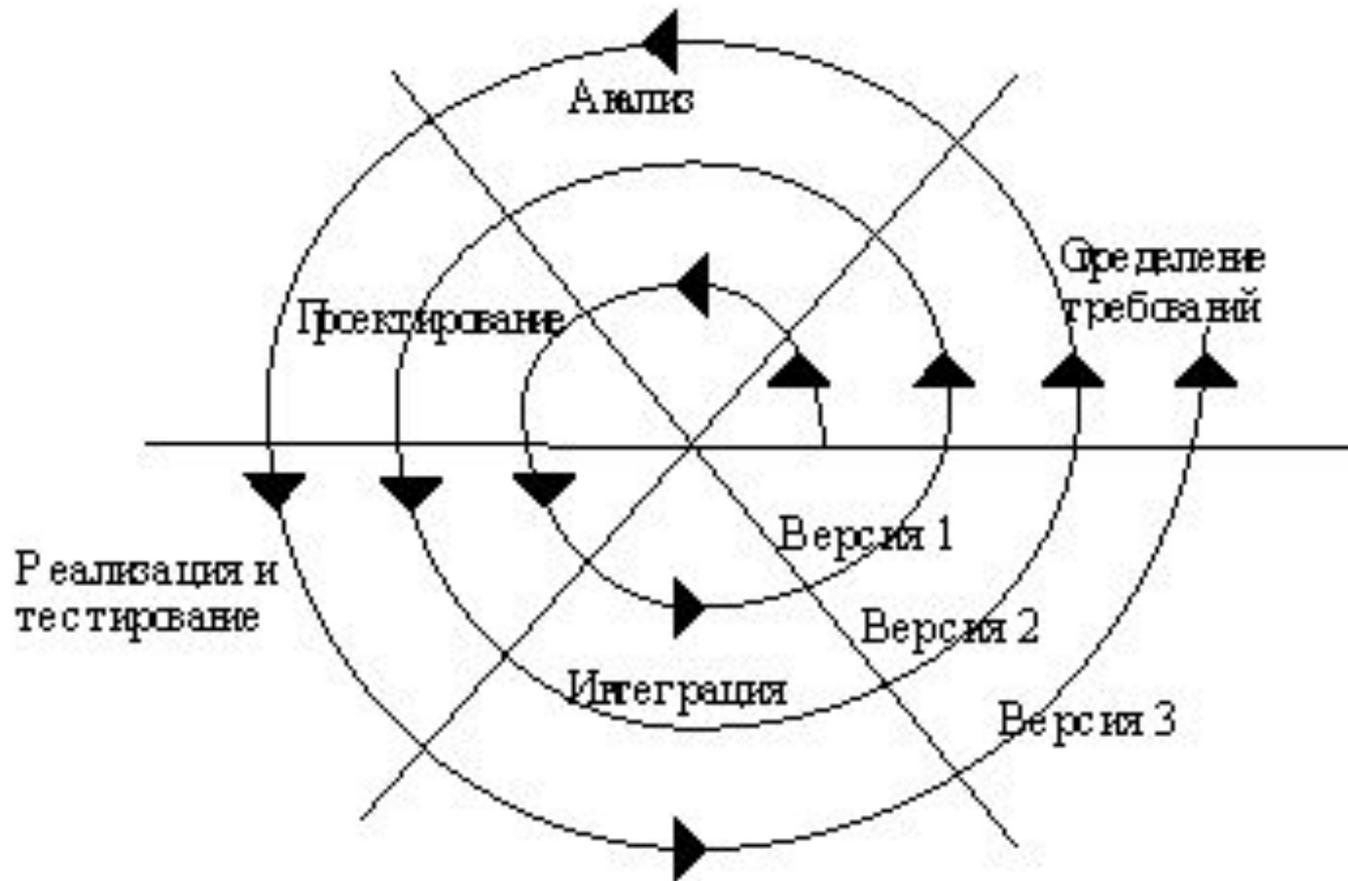
# Многопроходная модель

*Быстро создается работающая система.*

*Уменьшается возможность внесения изменений в процессе разработки.*

*Невозможен переход от текущей реализации к новой версии в течение построения текущей частичной реализации*

# Спиральная модель



# Технологии и инструментальные средства моделирования бизнес-процессов

**Структурный анализ** – метод исследования системы, которое начинается с общего обзора и затем детализируется, приобретая иерархическую структуру со все большим числом уровней.

**Объектно-ориентированное моделирование** - подразумевает описание статической структуры системы в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Каждый объект обладает своим собственным поведением, моделирующим поведение объекта реального мира.

Программные средства: IDEF Designer, ERwin\BPwin, Oracl Designer, BPM Workbench, Aris, Rational Rose, Ramus, StarUML

# Стандарты моделирования IDEF

## Назначение семейства стандартов IDEF

Стандарты IDEF предназначены для разработки бизнес-моделей и представляют собой набор спецификаций языка описания бизнес-процессов.

IDEF-методика создавалась в США в рамках программы компьютеризации промышленности ICAM – Integrated Computer Aided Manufacturing. Название стандарта расшифровывается как Icam DEFinition.

# Стандарты моделирования IDEF

К семейству IDEF относятся следующие стандарты:

IDEF0 – методология функционального моделирования

IDEF1 – методология моделирования информационных потоков

IDEF1X – методология построения реляционных структур

IDEF2 – методология динамического моделирования развития систем

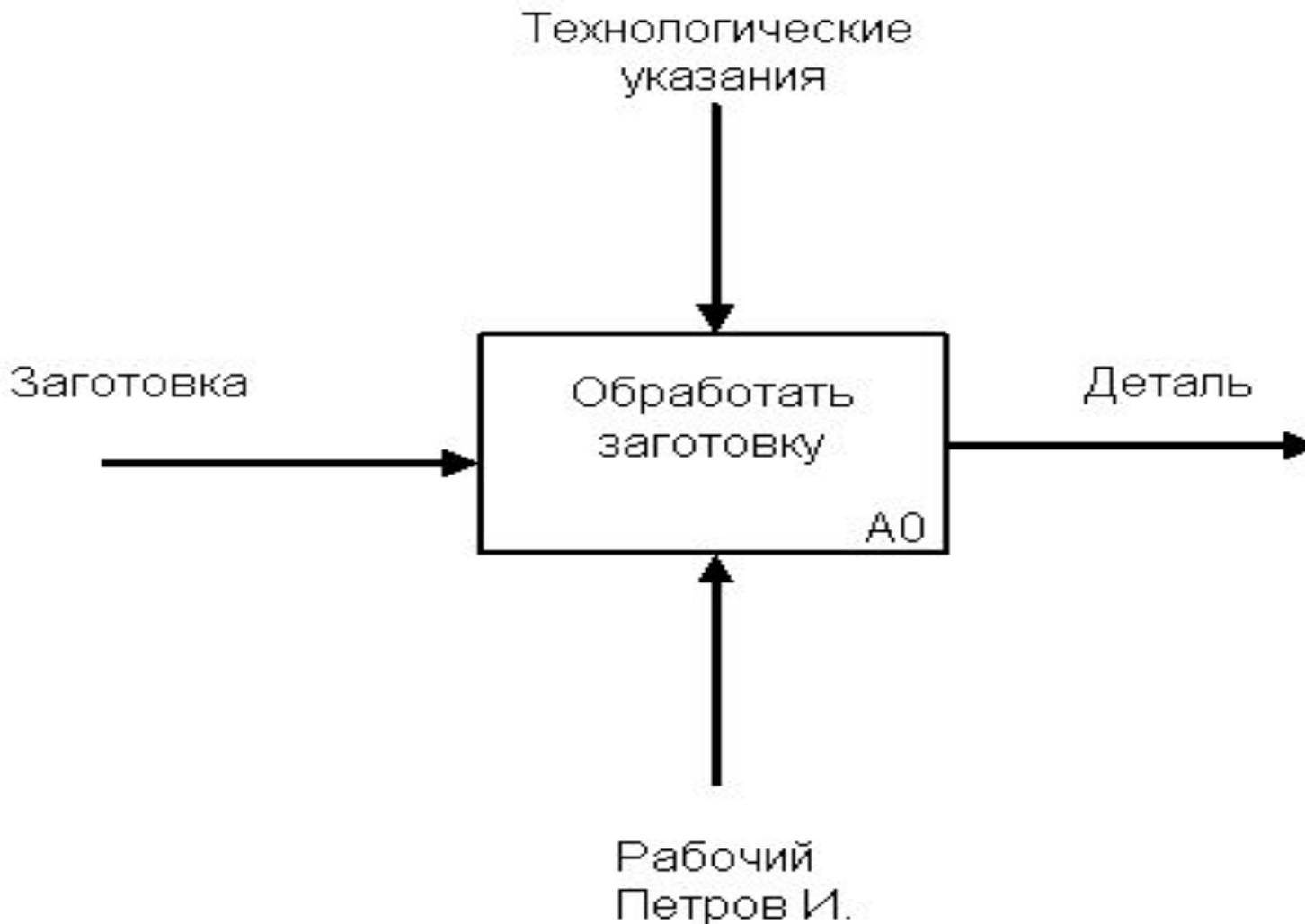
IDEF3 – методология документирования процессов в системе

IDEF4 – методология построения объектно-ориентированных систем

IDEF5 – методология онтологического описания сложных систем

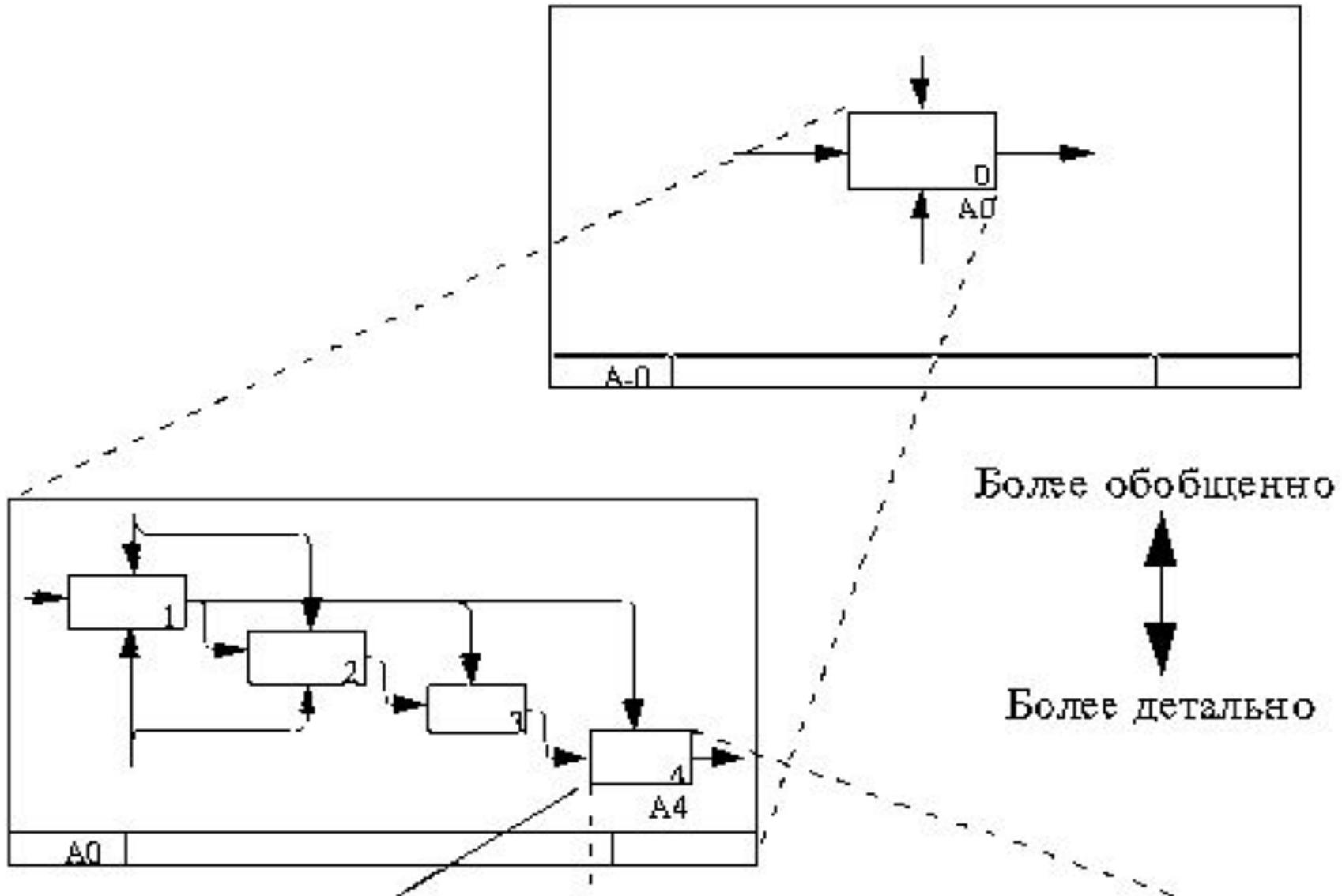
# Стандарты моделирования IDEF

Стандарт IDEF0 – функциональный блок



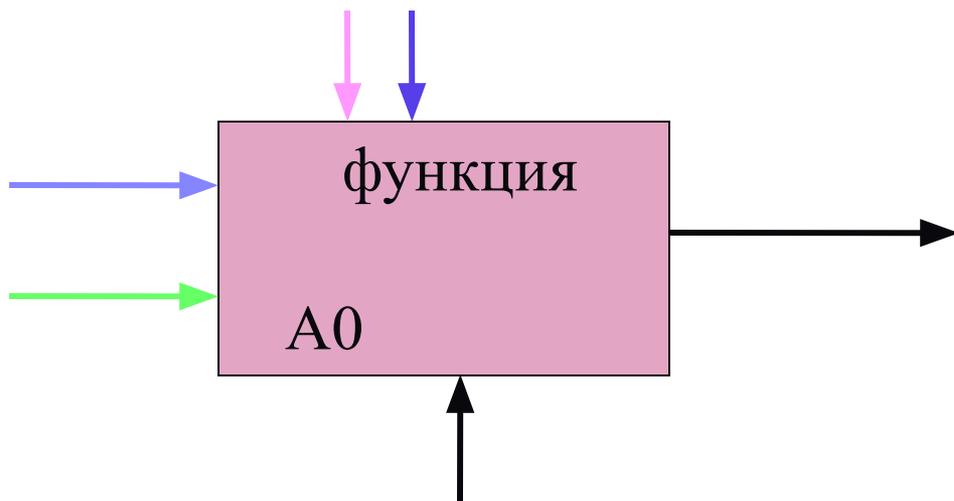
# Стандарты моделирования IDEF

## Стандарт IDEF0 – декомпозиция



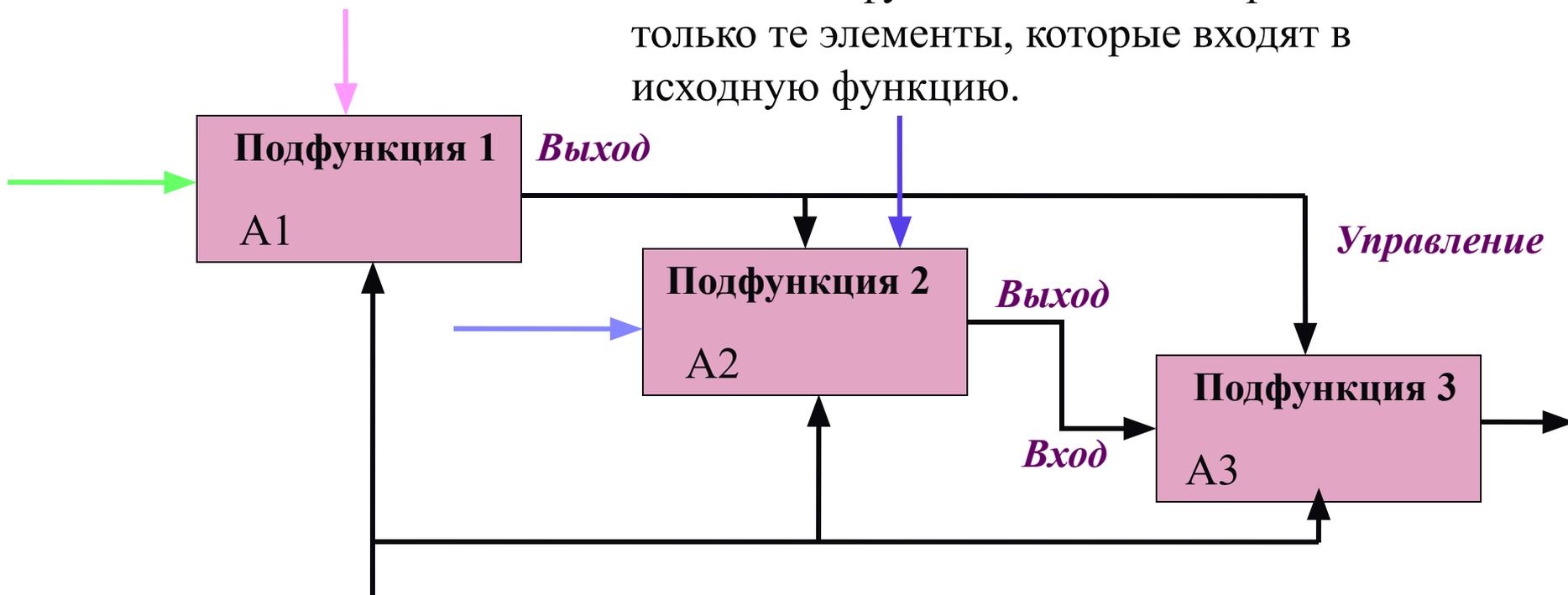
# Декомпозиция функциональных диаграмм

Контекстная диаграмма определяет все функции, входы и выходы, которые могут появиться на диаграммах нижних уровней



**IDEFO**

Каждая подфункция может содержать только те элементы, которые входят в исходную функцию.



# Стандарты моделирования IDEF.

## Два типа диаграмм в стандарте IDEF3

Существуют два типа диаграмм в стандарте IDEF3, представляющие описание одного и того же сценария технологического процесса в разных ракурсах.

Диаграммы относящиеся к первому типу называются диаграммами **Описания Последовательности Этапов Процесса (Process Flow Description Diagrams, PFDD)**.

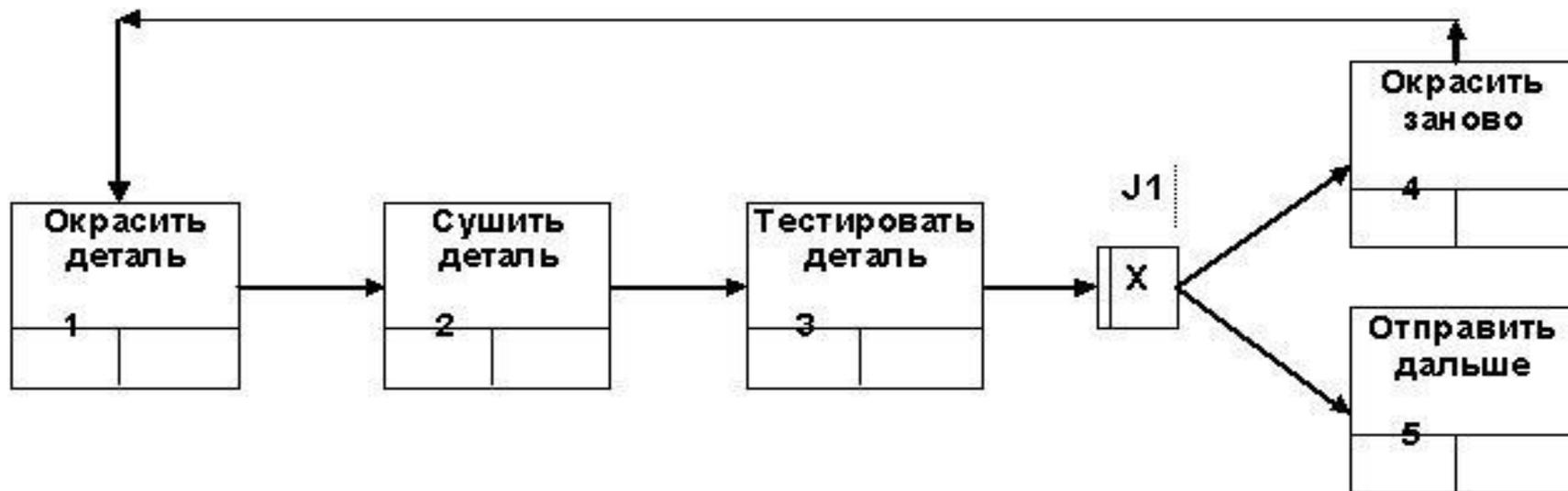


Рисунок 1. Пример PFDD диаграммы

# Стандарты моделирования IDEF.

## Два типа диаграмм в стандарте IDEF3

Второй - диаграммами Состояния Объекта в и его Трансформаций Процессе (Object State Transition Network, OSTN).

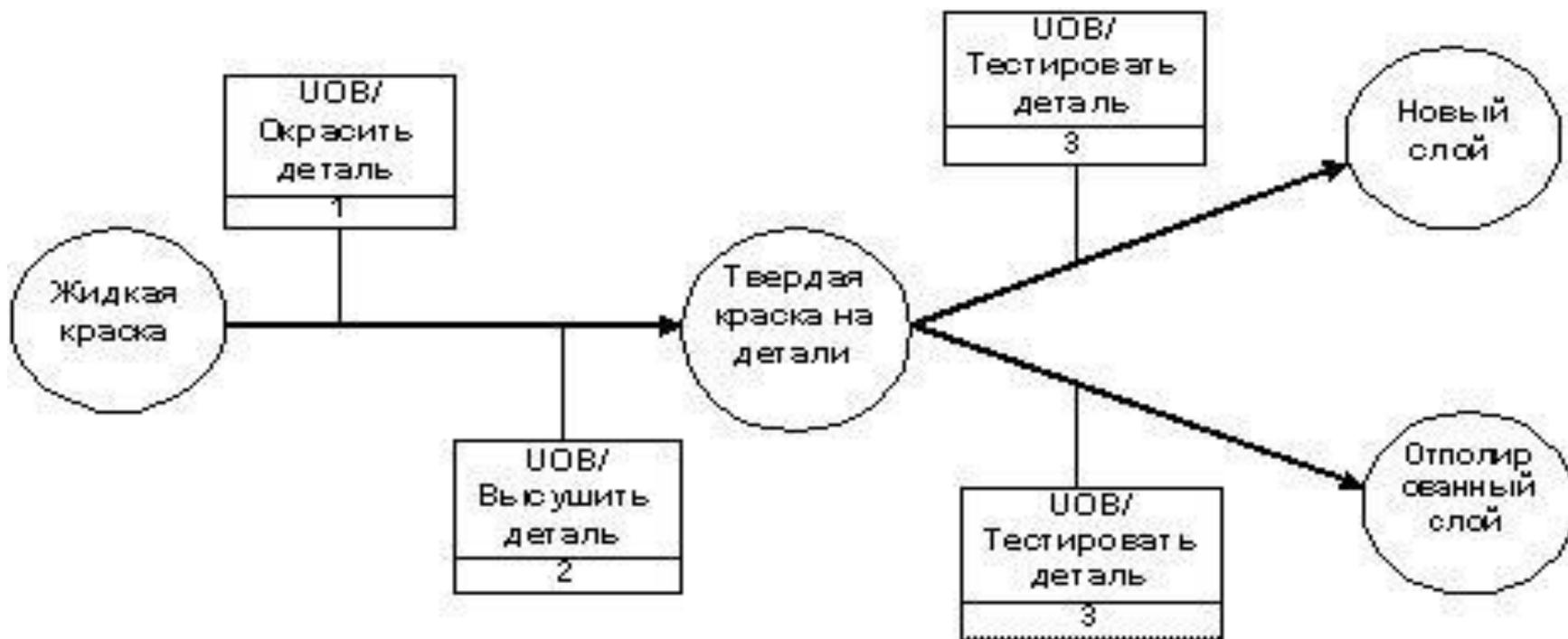
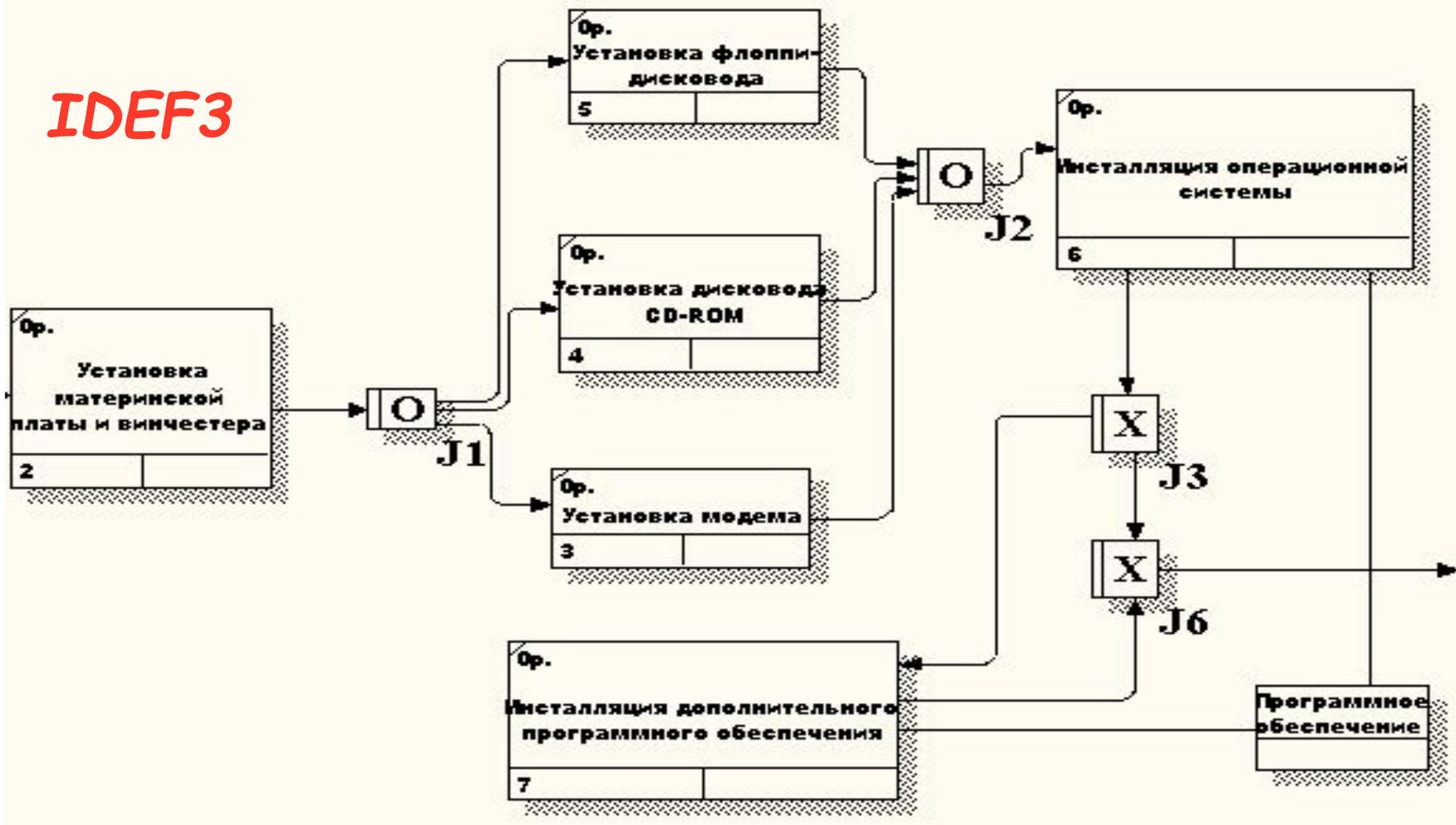


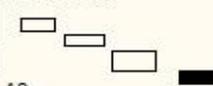
Рисунок 2. Пример OSTN диаграммы

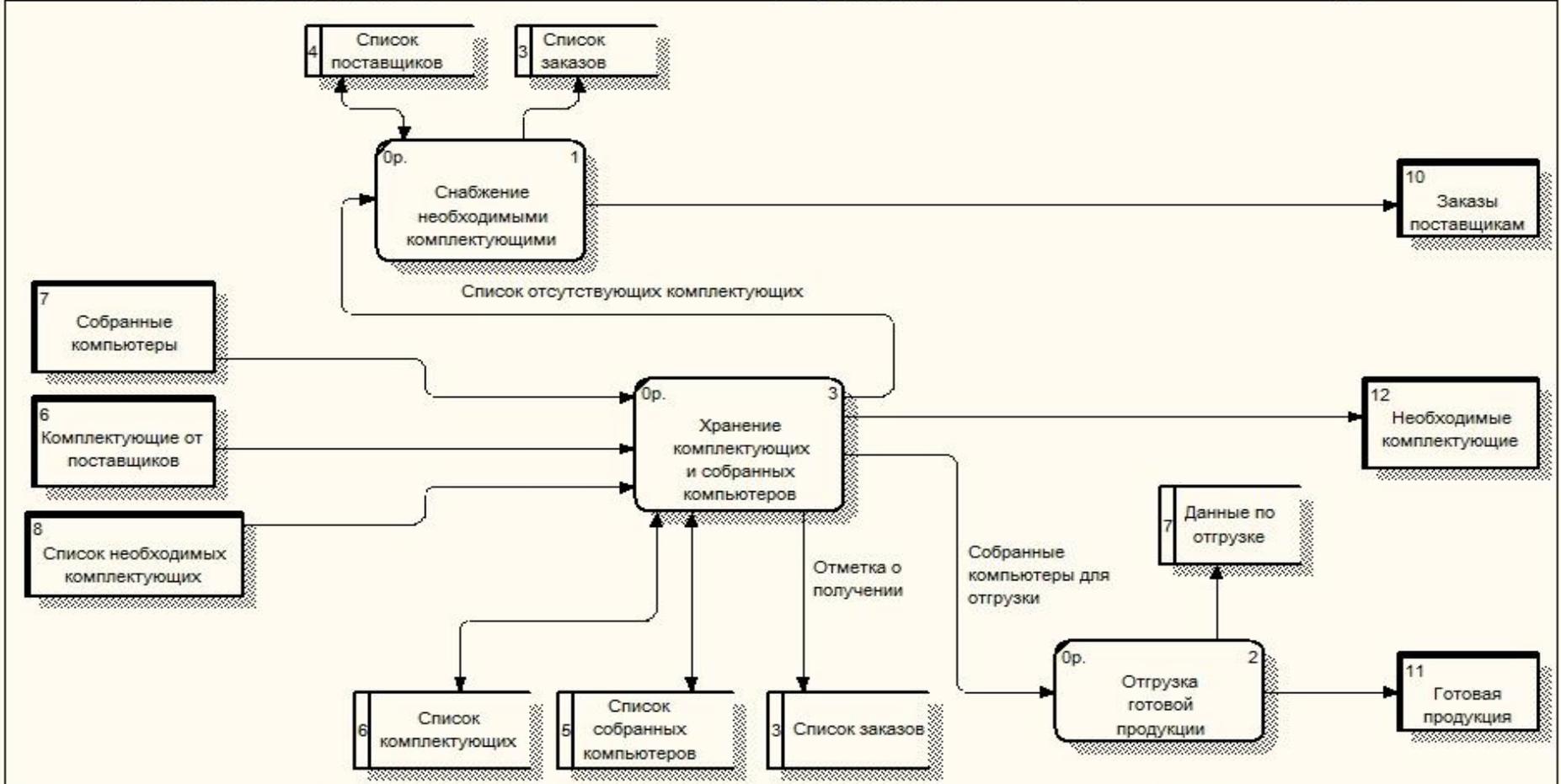
# Динамические аспекты поведения системы

**IDEF3**



# Модель потоков данных – диаграммы DFD (Data Flow Diagram)

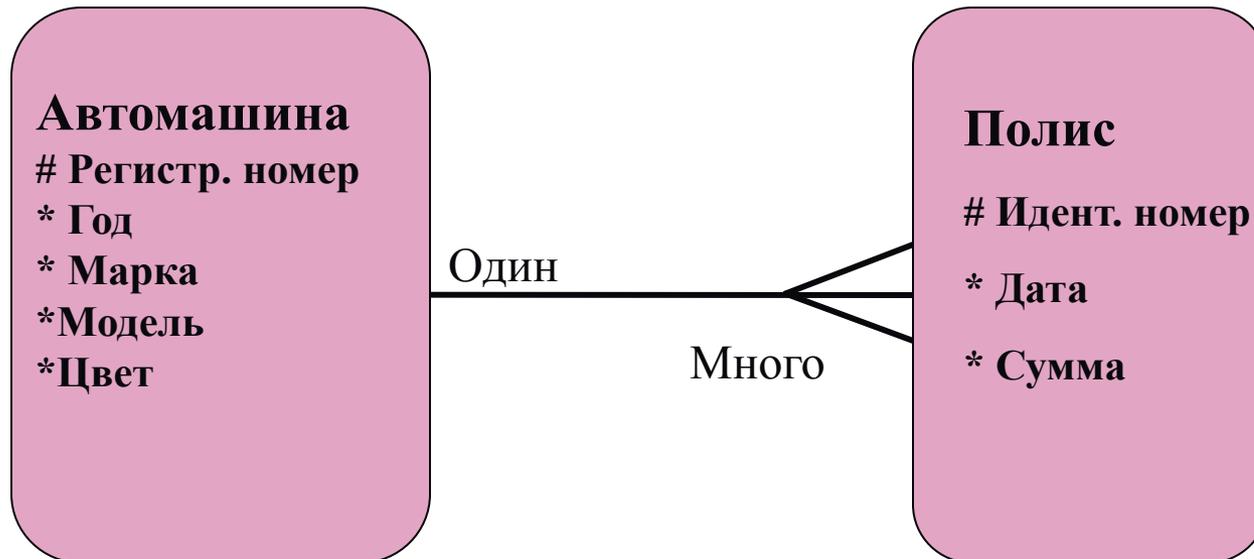
USED AT:	AUTHOR: Fastowsky G. Eduard	DATE: 12.04.2010	WORKING	READER	DATE	CONTEXT:  A0
	PROJECT: Computer Firm	REV: 13.04.2010	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						



NODE: <b>A4</b>	TITLE: <b>Отгрузка и снабжение</b>	NUMBER:
--------------------	---------------------------------------	---------

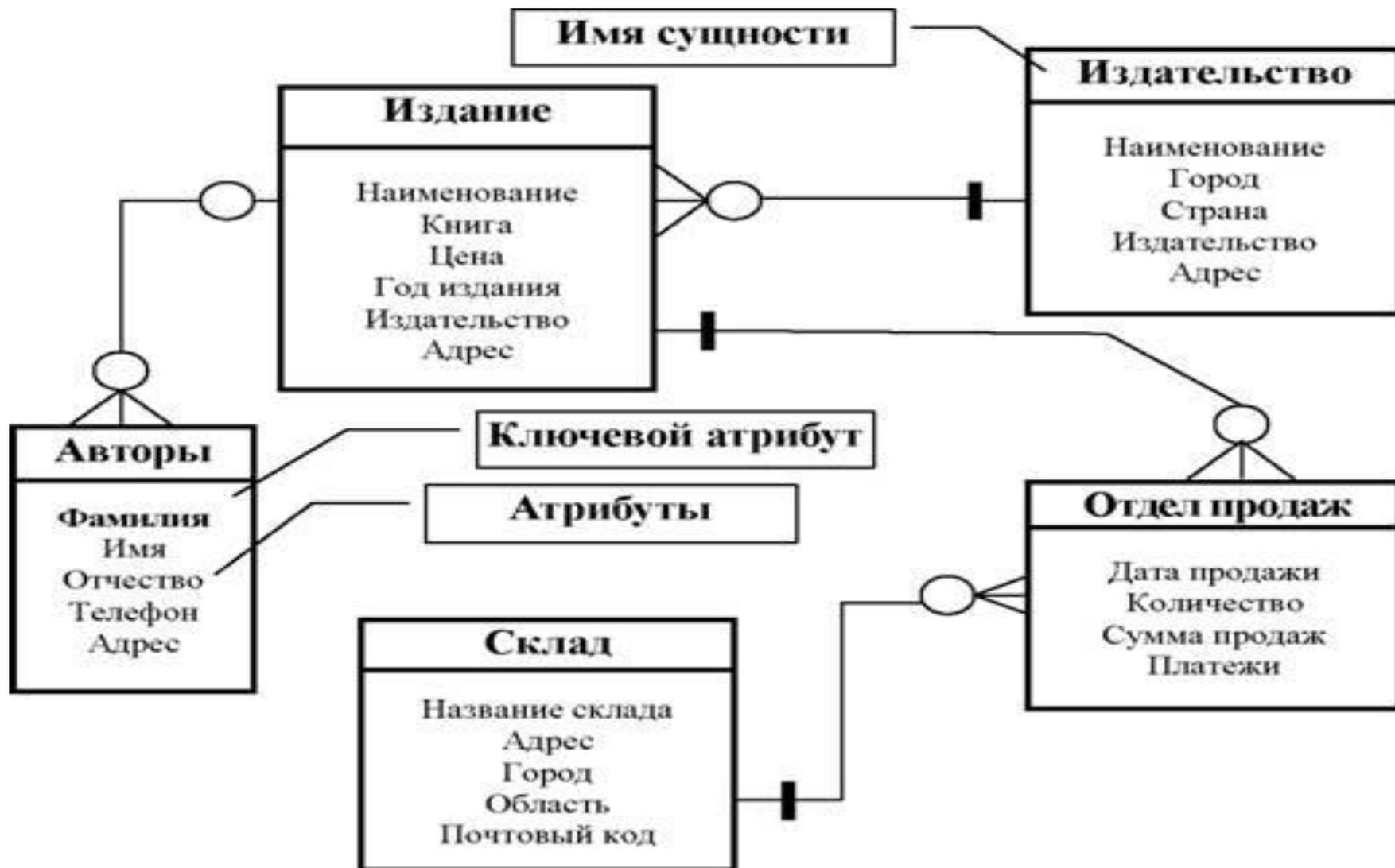
# Диаграммы ERD - «сущность-связь»

Описывают структуры данных, связанных с различными объектами модели; документируют сущности процесса (их идентификаторы, атрибуты) и способы взаимодействия между ними



**IDEF1X**

## Диаграммы ERD - «сущность-связь»



# UML (Unified Modeling Language)

- Унифицированный язык моделирования — язык графического описания для объектного моделирования при разработке ПО
- Язык широкого профиля, открытый стандарт графических обозначений для создания абстрактной модели системы
- UML не язык программирования, но из UML-моделей возможна генерация кода
- Основные авторы - Гради Буч, Джеймс Рамбо, Ивар Яacobсон
- Первая версия UML 1.0 - январь 1997 г.

# Виды диаграмм UML

## Structure Diagrams:

- Class diagram
- Component diagram
- Composite structure diagram
  - Collaboration (UML2.0)
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram (UML2.2)

## Behavior Diagrams:

- Activity diagram
- State Machine diagram
- Use case diagram
- **Interaction Diagrams:**
  - Communication diagram (UML2.0) / Collaboration (UML1.x)
  - Interaction overview diagram (UML2.0)
  - Sequence diagram
  - Timing diagram (UML2.0)

## Структурные диаграммы:

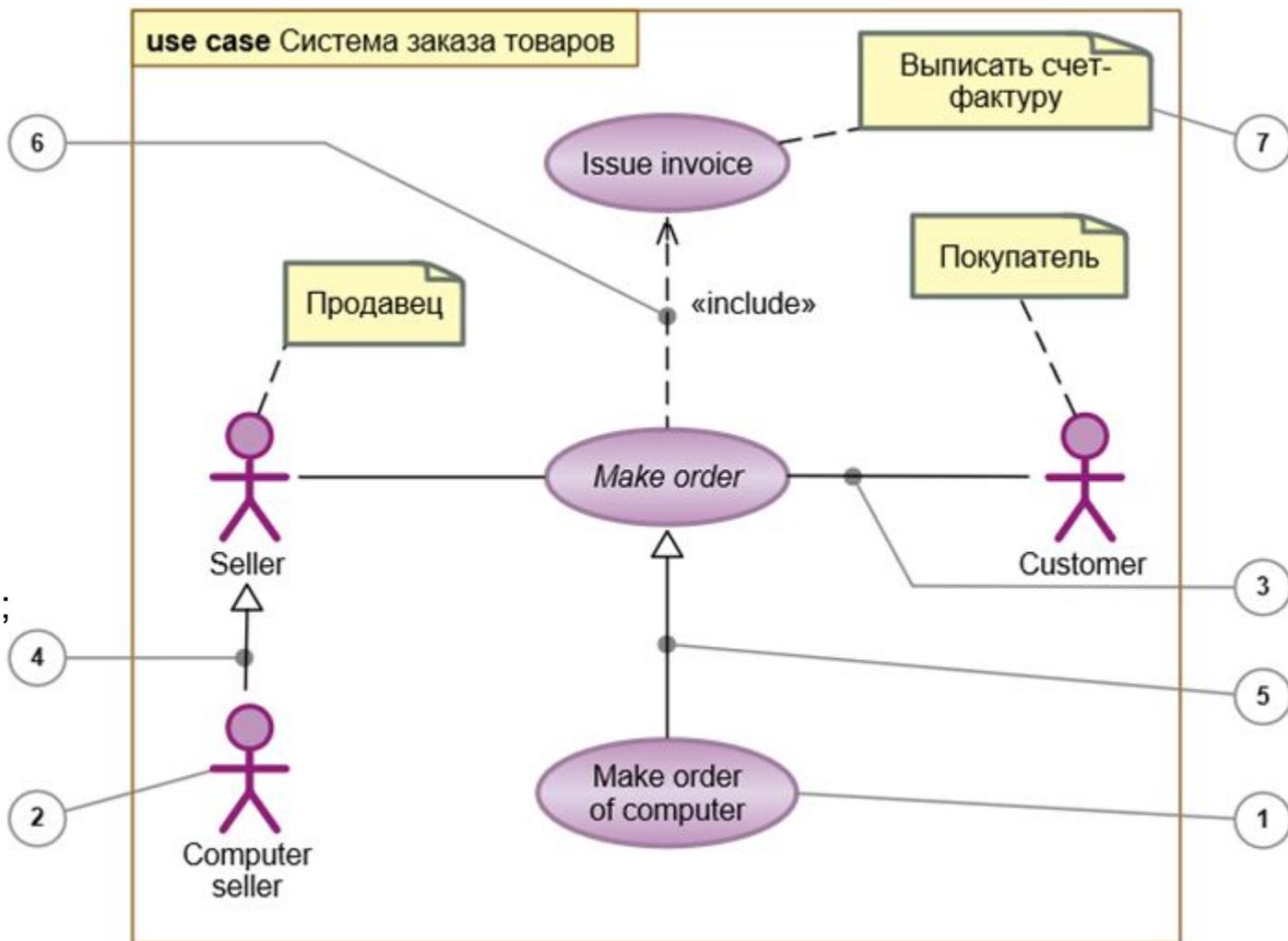
- Диаграмма классов
- Диаграмма компонентов
- Композитной/составной структуры
  - Диаграмма кооперации (UML2.0)
- Диаграмма развёртывания
- Диаграмма объектов
- Диаграмма пакетов
- Диаграмма профилей (UML2.2)

## Диаграммы поведения:

- Диаграмма деятельности
- Диаграмма состояний
- Диаграмма прецедентов
- **Диаграммы взаимодействия:**
  - Диаграмма коммуникации (UML2.0) / Диаграмма кооперации (UML1.x)
  - Диаграмма обзора взаимодействия (UML2.0)
  - Диаграмма последовательности
  - Диаграмма синхронизации (UML2.0)

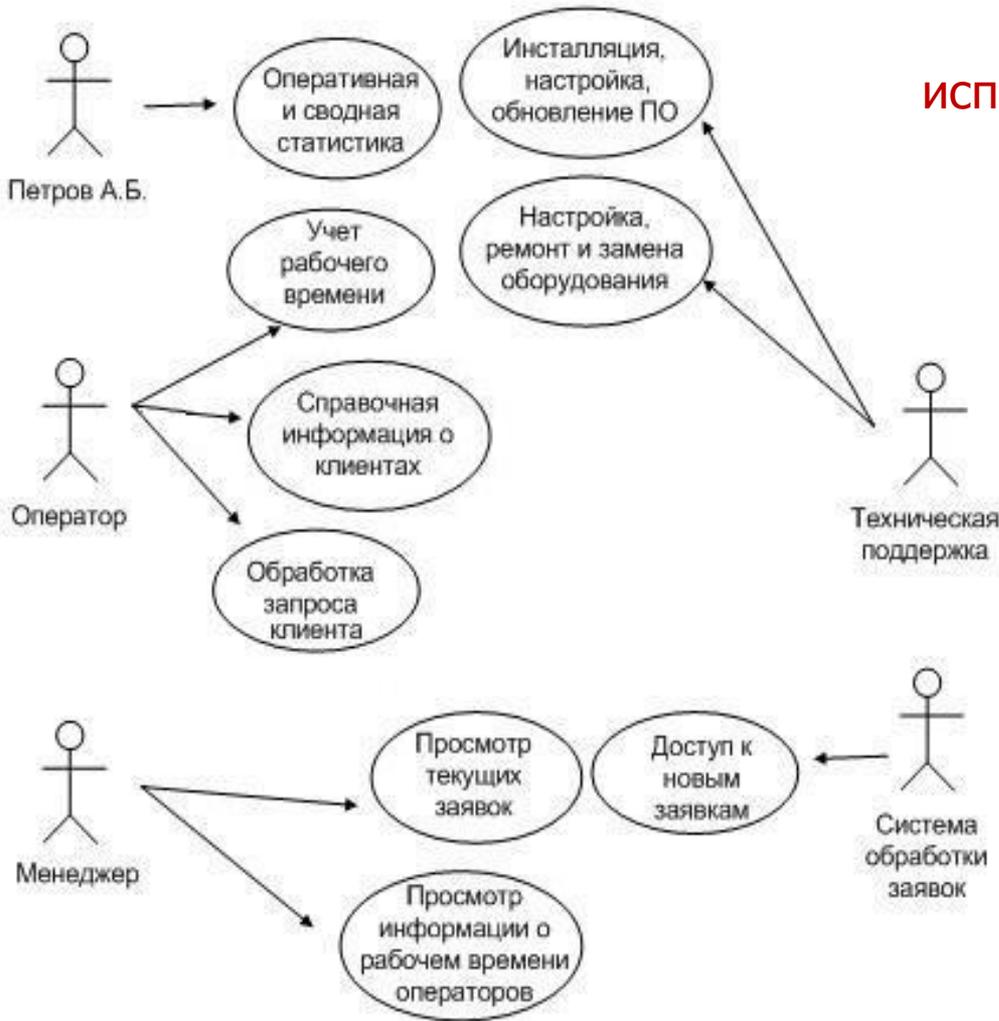
# Диаграмма случаев использования (прецедентов) (use case diagram)

- 1 - варианты использования;
- 2 - действующие лица;
- 7 – комментарии;
- Основные типы отношений:**
- 3 - ассоциация между действующим лицом и вариантом использования;
- 4 - обобщение между действующими лицами;
- 5 - обобщение между вариантами использования;
- 6 - зависимости (различных типов) между вариантами использования.



# Пример диаграммы случаев использования (прецедентов) (use case diagram)

диаграммы бизнес-случаев использования для системы обработки телефонных заявок

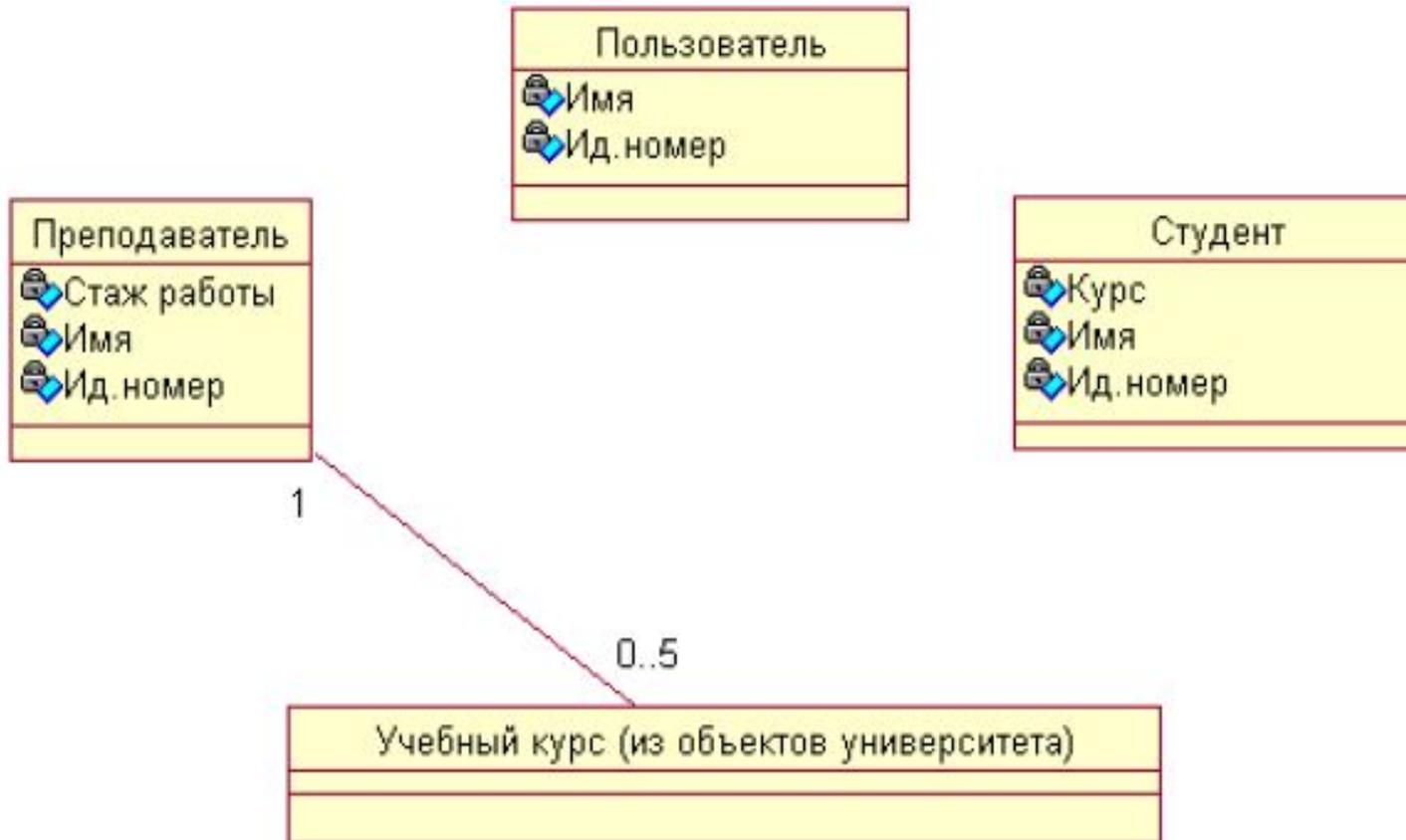


Основной задачей диаграмм случаев использования (прецедентов) является получение требований к системе от заказчика и пользователей

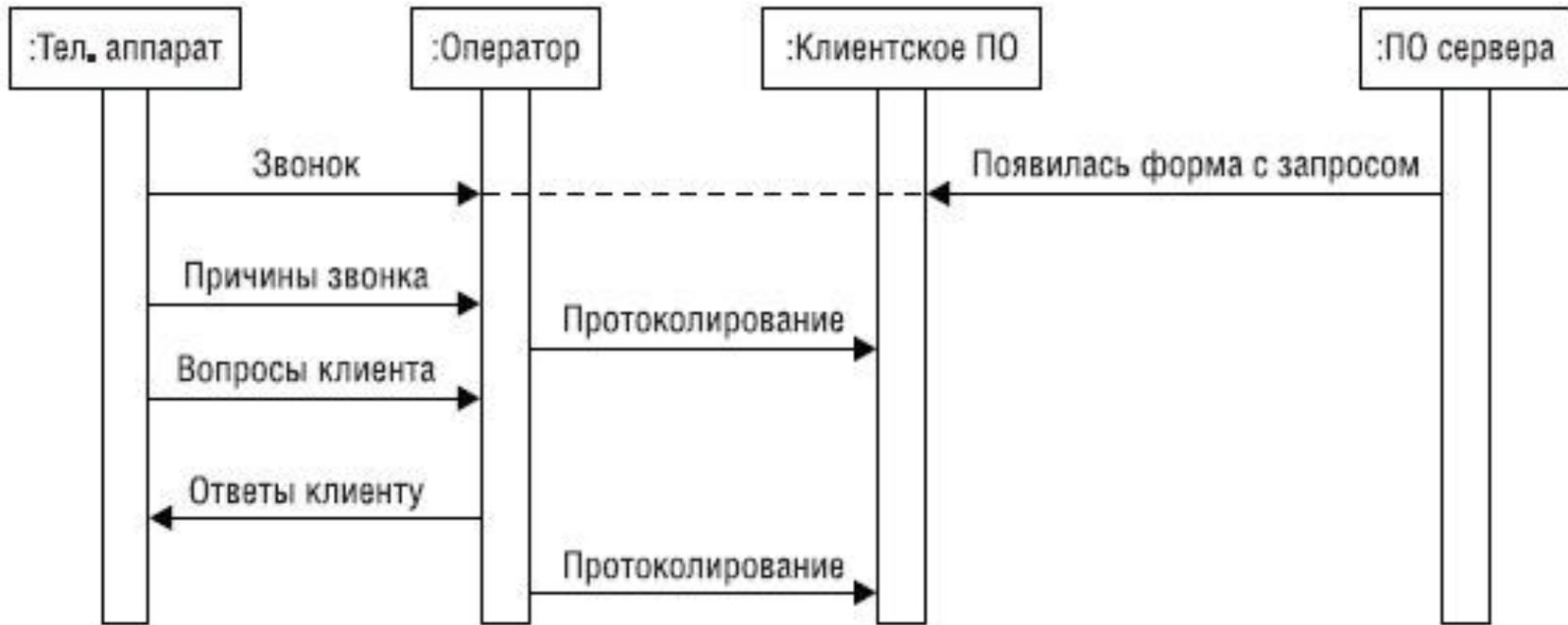
# Назначение диаграмм вариантов использования (use case diagram)

- Определить общие границы функциональности проектируемой системы в контексте моделируемой предметной области
- Специфицировать требования к функциональному поведению проектируемой системы в форме вариантов использования
- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями

# Примеры диаграммы классов в Rational Rose



# Диаграммы последовательностей (sequence diagrams)



На **диаграммах последовательностей**, так же как и на диаграммах коммуникаций, показываются **роли классов**

Фактически, на обеих диаграммах представлена одна и та же *информация*, но в разных видах. На диаграммах последовательностей она показана с точки зрения временного аспекта, на диаграммах коммуникаций – с точки зрения отношений взаимодействующих частей (то есть здесь яснее выражен структурный аспект)

Можно сказать, что *диаграмма последовательностей* является двойником диаграмм коммуникаций

# Диаграмма деятельности (activity diagram)

С их помощью удобно изображать бизнес-процессы - алгоритмы, по которым работает компания. Именно в эти алгоритмы должна встроиться информационная система, автоматизировав некоторую их часть

*Пример: бизнес-процесс по телефонной обработке заявок - общая схема работы оператора с клиентом*

